CLAIMS

We claim:

1. In a computerized system comprising a switching layer, and a sequence of one or more intermediate software layers of a network protocol stack, each of the intermediate software layers having a state object, a method for transferring control between one or more destination component devices and one or more source component devices, the control needed to process a plurality of state objects while still maintaining integrity of established network communication, the method comprising the following:

an act of generating an offload data structure, the offload data structure comprising a hierarchy of a plurality of state objects, the plurality of state objects corresponding to a network protocol state for one or more intermediate software layers;

an act of transferring from a source component device to a destination component device two or more state objects in the same intermediate software layer of the offload data structure; and

an act of the destination component processing the two or more state objects at the same protocol layer after the transfer.

2. The method of claim 1, wherein one of the destination component and source component is a peripheral device.

3. The method of claim 2, wherein at least one of the two or more state objects includes a cached state variable.

4. The method of claim 2, wherein the offload data structure includes a block list having at least a next block pointer that points to the same intermediate software layer of a different connection path through the network protocol stack, and a dependent block pointer that points to a different hierarchal layer of the same connection path through the network protocol stack.

WORKMAN NYDEGGER
A PROFESSIONAL CORPORATION
ATTORNEYS AT LAW
1000 EAGLE GATE TOWER
60 EAST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84111

5. The method of claim 2, wherein the act of transferring the offload data structure and the portion of the state object for each software layer is performed during the slow-start phase of a TCP connection.

6. The method of claim 2, further comprising an act of inserting an intermediate driver data structure in the offload data structure, the intermediate driver capable of deciphering and encapsulating incoming and outgoing data packets, as appropriate, wherein the outgoing data packets are packaged with one or more virtual local area network identifiers, and incoming packets are stripped of one or more virtual local area network identifiers.

7. The method of claim 6, wherein the intermediate driver is capable of securing network traffic over one or more network links using an IPSEC protocol by performing one or more of adding and removing an authentication header to a data packet, and encrypting and decrypting the data packet, as appropriate.

8. The method of claim 2, further comprising:

an act of the peripheral device generating one or more handles for one or more intermediate software layers, wherein the peripheral device transfers the one or more handles to one or more corresponding host protocol stack layers; and

an act of, if at least one of the intermediate software layers changes cached state, the at least one of the intermediate software layers updating the cached state of the peripheral device..

9. The method of claim 2, wherein a failover event occurs, further comprising:

an act of detecting that one or more links being processed at one or more corresponding peripheral devices have failed, the one or more failed peripheral devices having been given processor control of one or more offload data structures and one or more state objects; and

an act of detecting a different one or more peripheral devices that are capable of handling processing control of the one or more links by receiving the one

WORKMAN NYDEGGER
A PROFESSIONAL CORPORATION
ATTORNEYS AT LAW
1000 EAGLE GATE TOWER
60 EAST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84111

or more offload data structures and control of the one or more state objects.

10. The method of claim 9, further comprising transferring to the source component the one or more offload data structures and processing control of the one or more state objects from the one or more failed peripheral devices, thereby uploading processing control of the one or more links from the one or more failed peripheral devices to the source component.

11. The method of claim 10, further comprising an act of detecting a processing resource of the different one or more peripheral devices; and adjusting the one or more state objects that have been uploaded from the one or more failed peripheral devices to match the detected resource of the different one or more peripheral devices, so that the one or more links can be processed at the different one or more peripheral devices rather than at the failed one or more peripheral devices.

12. The method of claim 11, wherein the one or more links are aggregated 802.3ad links.

13. The method of claim 9, further comprising:

an act of detecting a processing resource of the different one or more peripheral devices;

if one of the one or more offloaded state objects can be processed by the different one or more peripheral devices based on the detected processing resource, transferring processing control of the one of the one or more offloaded state objects to the different one or more peripheral devices; and

if the one or more offloaded state objects cannot be processed by the different one or more peripheral devices based on the detected processing resource, transferring processing control of the one of the one or more offloaded state objects to the host protocol stack.

WORKMAN NYDEGGER
A PROFESSIONAL CORPORATION
ATTORNEYS AT LAW
1000 EAGLE GATE TOWER
60 EAST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84111

14. The method of claim 13, wherein the one or more links are aggregated 802.3ad links.

WORKMAN NYDEGGER
A PROFESSIONAL CORPORATION
ATTORNEYS AT LAW
1000 EAGLE GATE TOWER
60 EAST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84111

15. In a computerized system comprising a switching layer, and a sequence of one or more intermediate software layers of a network protocol stack, each of the intermediate software layers having a state object, a method for transferring control between one or more destination component devices and one or more source component devices, the control needed to process a plurality of state objects while still maintaining integrity of established network communication, the method comprising the following:

a step for reducing the processing demands of a computerized system that is processing two or more state objects corresponding to the same intermediate software layer by transferring processing control of the two or more state objects in a way that preserves the integrity of network communications with regard to state changes, failover events, and VLAN tags; and

an act of the destination component processing the two or more state objects at the same protocol layer after the transfer.

16. The method of claim 15, wherein the step for transferring includes:

an act of generating an offload data structure, the offload data structure comprising a hierarchy of a plurality of state objects, the plurality of state objects corresponding to a network protocol state for one or more intermediate software layers;

an act of transferring from a source component device to a destination component device two or more state objects in the same intermediate software layer of the offload data structure.

WORKMAN NYDEGGER
A PROFESSIONAL CORPORATION
ATTORNEYS AT LAW
1000 EAGLE GATE TOWER
60 EAST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84111

17. The method of claim 16, further comprising a step for updating the destination component in the event of a change in the state object, or in the event of a change in the offload data structure so that the destination component can still process the two or more state objects as intended without losing network communication integrity.

18. The method of claim 17, wherein the step for updating the destination component includes the following:

an act of the peripheral device generating one or more handles for one or more intermediate software layers, wherein the peripheral device transfers the one or more handles to one or more corresponding host protocol stack layers; and

an act of, if at least one of the intermediate software layers changes cached state, the at least one of the intermediate software layers updating the cached state of the peripheral device.

19. The method of claim 15, further comprising a step for dynamically preserving the integrity of the network communications in the event of a failover event so that the one or more aggregated network links can be processed appropriately at a different one or more peripheral devices.

20. The method of claim 19, wherein the step for dynamically preserving the integrity of the two or more network links includes:

an act of detecting that one or more links being processed at one or more corresponding peripheral devices have failed, the one or more failed peripheral devices having been given processor control of one or more offload data structures and one or more state objects; and

an act of detecting a different one or more peripheral devices that are capable of handling processing control of the one or more links by receiving the one or more offload data structures and control of the one or more state objects;

an act of detecting a processing resource of the different one or more peripheral devices;

WORKMAN NYDEGGER
A PROFESSIONAL CORPORATION
ATTORNEYS AT LAW
1000 EAGLE GATE TOWER
60 EAST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84111

if one of the one or more offloaded state objects can be processed by the different one or more peripheral devices based on the detected processing resource, transferring processing control of the one of the one or more offloaded state objects to the different one or more peripheral devices; and

if the one or more offloaded state objects cannot be processed by the different one or more peripheral devices based on the detected processing resource, transferring processing control of the one of the one or more offloaded state objects to the host protocol stack.

WORKMAN NYDEGGER
A PROFESSIONAL CORPORATION
ATTORNEYS AT LAW
1000 EAGLE GATE TOWER
60 EAST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84111

21. In a computerized system comprising a switching layer, and a sequence of one or more intermediate software layers of a network protocol stack, each of the intermediate software layers having a state object, a method for transferring control between one or more destination component devices and one or more source component devices, the control needed to process a plurality of state objects while still maintaining integrity of established network communication, the method comprising the following:

an act of generating an offload data structure, the offload data structure comprising a plurality of short-lived network connection,;

an act of transferring from a source component device to a destination component device two or more short-lived network connections in the same intermediate software layer of the offload data structure; and

an act of the destination component processing the two or more short-lived connections after the transfer.

22. The method of claim 21, wherein one of the destination component and source component is a peripheral device.

23. The method of claim 22, wherein a state object includes a cached state variable, and one or more of a constant state variable, and a delegated state variable.

24. The method of claim 22, wherein the offload data structure includes a block list having at least a next block pointer that points to the same intermediate software layer of a different connection path through the network protocol stack, and a dependent block pointer that points to a different hierarchal layer of the same connection path through the network protocol stack.

25. The method of claim 22, wherein the act of transferring the two or more short-lived connections is performed during the slow-start phase of a TCP connection

WORKMAN NYDEGGER
A PROFESSIONAL CORPORATION
ATTORNEYS AT LAW
1000 EAGLE GATE TOWER
60 EAST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84111

26. The method of claim 22, further comprising an act of inserting an intermediate driver data structure in the offload data structure, the intermediate driver capable of deciphering and encapsulating incoming and outgoing data packets, as appropriate, wherein the outgoing data packets are packaged with one or more virtual local area network identifiers and incoming packets are stripped of one or more virtual local area network identifiers.

27. The method of claim 26, wherein the intermediate driver is capable of securing network traffic over one or more network links using an IPSEC protocol by performing one or more of adding and removing an authentication header to a data packet, and encrypting and decrypting the data packet, as appropriate.

28. The method of claim 22, further comprising:

an act of generating a handle for each layer of the intermediate software layers, wherein the peripheral device transfers the handle for each intermediate software layer to each intermediate software layer; and

an act of, if at least one of the intermediate software layers changes cached state, the at least one of the intermediate software layers updating the cached state of the peripheral device.

29. The method of claim 22, further comprising:

an act of detecting that one or more links being processed at one or more corresponding peripheral devices have failed, the one or more failed peripheral devices having been given processor control of the two or more short-lived network connections; and

an act of detecting a different one or more peripheral devices that are capable of handling processing control of the one or more links by receiving the two or more short-lived network connections.

WORKMAN NYDEGGER
A PROFESSIONAL CORPORATION
ATTORNEYS AT LAW
1000 EAGLE GATE TOWER
60 EAST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84111

30. The method of claim 29, further comprising transferring to the source component the one or more offload data structures and processing control of the one or more state objects from the one or more failed peripheral devices, thereby uploading processing control of the one or more links from the one or more failed peripheral devices to the source component.

31. The method of claim 30, further comprising an act of detecting a processing resource of the different one or more peripheral devices; and adjusting the one or more state objects that have been uploaded from the one or more failed peripheral devices to match the detected resource of the different one or more peripheral devices, so that the one or more links can be processed at the different one or more peripheral devices rather than at the failed one or more peripheral devices.

32. The method of claim 31, wherein the one or more links are aggregated 802.3ad links.

33. The method of claim 29, further comprising:

an act of detecting a processing resource of the different one or more peripheral devices;

if one of the one or more offloaded state objects can be processed by the different one or more peripheral devices based on the detected processing resource, transferring processing control of the one of the one or more offloaded state objects to the different one or more peripheral devices; and

if the one or more offloaded state objects cannot be processed by the different one or more peripheral devices based on the detected processing resource, transferring processing control of the one of the one or more offloaded state objects to the host protocol stack.

34. The method of claim 33, wherein the one or more links are aggregated 802.3ad links.

WORKMAN NYDEGGER
A PROFESSIONAL CORPORATION
ATTORNEYS AT LAW
1000 EAGLE GATE TOWER
60 EAST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84111

35. In a computerized system comprising a central processing unit, a plurality of peripheral devices, a switching layer, and a sequence of one or more intermediate software layers, a method for transferring a plurality of offloaded aggregated state objects from one or more of the peripheral devices that has failed to another peripheral device that is capable of receiving one or more of the state objects in the plurality of offloaded state objects, while still maintaining integrity of each of the established network communication, the method comprising the following:

an act of detecting that one or more links being processed at one or more peripheral devices has failed, the one or more failed peripheral devices having been given processing control of one or more state objects;

an act of detecting a different one or more peripheral devices that are capable of handling processing control of only the one or more links;

an act of detecting a different one or more peripheral devices that are capable of receiving processing control of the one or more links and one or more state objects; and

an act of detecting a processing resource of any of the detected one or more peripheral devices.

36. The method of claim 35, further comprising:

if the different one or more peripheral devices are capable of handling processing control of only the one or more links based on the detected processing resource, an act of transferring processing control of only the one or more links to the different one or more peripheral devices;

if the different one or more peripheral devices are capable of handling processing control of one or more links and one or more state objects based on the detected processing resource, an act of transferring processing control of the one or more links and the one or more offloaded state objects to the different one or more peripheral devices; and

WORKMAN NYDEGGER
A PROFESSIONAL CORPORATION
ATTORNEYS AT LAW
1000 EAGLE GATE TOWER
60 EAST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84111

an act of transferring processing control of any remaining of the one or more links and any remaining of the one or more offloaded state objects to the host protocol stack.

37. The method of claim 36, wherein one or more of the offloaded state objects include at least a cached state variable.

38. The method of claim 36, wherein the one or more links are aggregated 802.3ad links.

39. The method of claim 36, further comprising, prior to the act of transferring processing control of the one or more links and the one or more offloaded state objects, inserting an intermediate driver adjacent a data structure comprising the offloaded state objects, the intermediate driver capable of performing at least one of multiplexing one or more VLAN tags, de-multiplexing one or more VLAN tags, directing network state object traffic over one or more peripheral devices, and securing network data packets using an IPSEC protocol.

WORKMAN NYDEGGER
A PROFESSIONAL CORPORATION
ATTORNEYS AT LAW
1000 EAGLE GATE TOWER
60 EAST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84111

40. In a computerized system comprising a host processing unit, one or more peripheral devices, a switching layer, and a sequence of one or more intermediate software layers at least some of which having a state object, a method of inserting an intermediate driver adjacent an offload data structure in order to better facilitate, during a network connection offload and/ or upload process, a virtual local area network, the method comprising the following:

an act of generating an offload data structure, the offload data structure comprising a hierarchy of a plurality of state objects, the plurality of state objects corresponding to a network protocol state for one or more intermediate software layers;

an act of inserting one or more intermediate driver data structures within the offload data structure; and

an act of transferring from a source component device to one or more peripheral devices one or more state objects of a corresponding intermediate software layer of the offload data structure.

41. The method of claim 40, wherein a failover event occurs, an intermediate driver corresponding with at least one of the one or more intermediate driver data structures performs a method comprising:

an act of detecting a different one or more peripheral devices that are capable of handling processing control of only the one or more links;

an act of detecting a different one or more peripheral devices that are capable of receiving processing control of the one or more links and one or more state objects; and

an act of detecting a processing resource of any of the detected one or more peripheral devices.

WORKMAN NYDEGGER
A PROFESSIONAL CORPORATION
ATTORNEYS AT LAW
1000 EAGLE GATE TOWER
60 EAST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84111

42. The method of claim 40, wherein the one or more intermediate driver data structures are inserted into the offload data structure prior to offloading processing control of the offload data structure to the one or more peripheral devices.

43. The method of claim 40, wherein the one or more intermediate driver data structures are inserted at the point of initiating an offload request of the offload data structure to the one or more peripheral devices.

44. The method of claim 43, wherein, at least one of the one or more intermediate drivers includes instructions for performing an act of directing the sub-data packet to a VLAN address based on a virtual LAN tag associated with the sub-data packet, the instructions further comprising and at least one of:

an act of receiving a multiplexed data packet;

an act of de-multiplexing the data packet into one or more sub-data packets;

an act of multiplexing a data packet; and

an act of sending a multiplexed data packet.

45. The method of claim 40, wherein the offload data structure corresponds with two or more state objects corresponding to the same intermediate software layer, and wherein the two or more state objects point to a single state object at a lower software layer of a network protocol stack, wherein the offload data structure has the appearance of an inverted tree.

46. The method of claim 45, further comprising an act of transferring processing control of the entire inverted tree offload data structure to one or more of the one or more peripheral devices.

47. The method of claim 46, further comprising an act of transferring processing control of one or more of the state objects of the inverted tree offload data structure from the at least one of the one or more peripheral devices to the host protocol stack.

48. The method of claim 47, wherein the act of transferring from a source component device to one or more peripheral devices one or more state objects occurs in response to an act of detecting that at least one of the one or more peripheral devices has failed.

WORKMAN NYDEGGER
A PROFESSIONAL CORPORATION
ATTORNEYS AT LAW
1000 EAGLE GATE TOWER
60 EAST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84111

49. The method of claim 48, wherein an intermediate driver that corresponds with at least one of the one or more intermediate driver data structures performs the method comprising:

an act of detecting a different one or more peripheral devices that are capable of handling processing control of only the one or more links;

an act of detecting a different one or more peripheral devices that are capable of receiving processing control of the one or more links and one or more state objects; and

an act of detecting a processing resource of any of the detected one or more peripheral devices.

if the different one or more peripheral devices are capable of handling processing control of only the one or more links based on the detected processing resource, an act of transferring processing control of only the one or more links to the different one or more peripheral devices;

if the different one or more peripheral devices are capable of handling processing control of one or more links and one or more state objects based on the detected processing resource, an act of transferring processing control of the one or more links and the one or more offloaded state objects to the different one or more peripheral devices; and

an act of transferring processing control of any remaining of the one or more links and any remaining of the one or more offloaded state objects to the host protocol stack.

WORKMAN NYDEGGER
A PROFESSIONAL CORPORATION
ATTORNEYS AT LAW
1000 EAGLE GATE TOWER
60 EAST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84111

50. A computer program product for use in a computerized system comprising a switching layer, and a sequence of one or more intermediate software layers of a network protocol stack, each of the intermediate software layers having a state object, the computer program product for implementing a method for transferring control between one or more destination component devices and one or more source component devices, the control needed to process a plurality of state objects while still maintaining integrity of established network communication, the method comprising the following, the computer program product comprising one or more computer-readable media having thereon computer-executable instructions that, when executed by one or more processors at the host computing system, cause the host computing system to perform the method, the method comprising the following:

an act of generating an offload data structure, the offload data structure comprising a hierarchy of a plurality of state objects, the plurality of state objects corresponding to a network protocol state for one or more intermediate software layers;

an act of transferring from a source component device to a destination component device two or more state objects in the same intermediate software layer of the offload data structure; and

an act of the destination component processing the two or more state objects at the same protocol layer after the transfer.

51. The computer program product of claim 50, wherein the computer-executable instructions, when executed by the one or more processors, further cause the computerized system to perform the following:

an act of generating a handle for each layer of the intermediate software layers, wherein the destination component is a NIC, and the NIC transfers the handle for each intermediate software layer to each intermediate software layer; and

an act of, if at least one of the intermediate software layers changes cached state, the at least one of the intermediate software layers updating the cached state of the NIC.

WORKMAN NYDEGGER
A PROFESSIONAL CORPORATION
ATTORNEYS AT LAW
1000 EAGLE GATE TOWER
60 EAST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84111

52. The computer program product of claim 50, wherein the computer-executable instructions, when executed by the one or more processors, further cause the computerized system to perform the following:

an act of detecting that one or more links being processed at one or more corresponding peripheral devices have failed, the one or more failed peripheral devices having been given processor control of one or more offload data structures and one or more state objects; and

an act of detecting a different one or more peripheral devices that are capable of handling processing control of the one or more links by receiving the one or more offload data structures and control of the one or more state objects.

53. The computer program product of claim 52, wherein the computer-executable instructions, when executed by the one or more processors, further cause the computerized system to perform the following:

an act of detecting a processing resource of the different one or more peripheral devices;

if one of the one or more offloaded state objects can be processed by the different one or more peripheral devices based on the detected processing resource, transferring processing control of the one of the one or more offloaded state objects to the different one or more peripheral devices; and

if the one or more offloaded state objects cannot be processed by the different one or more peripheral devices based on the detected processing resource, transferring processing control of the one of the one or more offloaded state objects to the host protocol stack.

WORKMAN NYDEGGER
A PROFESSIONAL CORPORATION
ATTORNEYS AT LAW
1000 EAGLE GATE TOWER
60 EAST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84111